Get Authentik logs into Wazuh.

?? Phase 1: Setup the Authentik LXC (The Log Source)

Target: The Proxmox LXC where Authentik is running.

1. Install & Configure the Log Receiver

Most LXCs don’t have a syslog service active. You need this to catch Docker logs.

```
# Install rsyslog
apt update && apt install rsyslog -y

# Enable UDP reception (Open /etc/rsyslog.conf)
# Uncomment these two lines:
# module(load="imudp")
# input(type="imudp" port="514")

# Restart to apply
systemctl restart rsyslog
```

2. Route Docker Logs to the Host

Update your `docker-compose.yml` for the `server` and `worker` services so they talk to the service you just enabled.

- **Path:** `/opt/authentik/docker-compose.yml` (or wherever your yaml is)
- **Action:** Add the `logging` block.

```
logging:
  driver: syslog
  options:
    syslog-address: "udp://127.0.0.1:514"
    tag: "authentik"
```

Apply the change:

```
docker compose up -d
```

3. Configure the Wazuh Agent

Tell the agent to watch the system log and give it permission to read it.

- **Path:** `/var/ossec/etc/ossec.conf`
- **Action:** Add the `<localfile>` block.

```
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/syslog</location>
</localfile>
```

Crucial Step for Permissions:

```
usermod -a -G adm wazuh
systemctl restart wazuh-agent
```

? Phase 2: Setup the Wazuh Manager (The Brain)

Target: The LXC/Server where your Wazuh Manager lives.

1. Create the Decoder

This tells Wazuh how to "break apart" the Authentik JSON log into fields like `user` and `srcip`.

- **Path:** **`/var/ossec/etc/decoders/local_decoder.xml`**

```
<decoder name="authentik">
  <program_name>authentik</program_name>
</decoder>

<decoder name="authentik-fields">
  <parent>authentik</parent>
```

```
  <regex>"event":\s*"([^"]+)",.*"remote":\s*"([^"]+)",.*"user":\s*"([^"]*)"</regex>
  <order>action, srcip, user</order>
</decoder>
```

2. Create the Security Rules

This tells Wazuh which logs are "important" enough to show on the dashboard.

- **Path:** `/var/ossec/etc/rules/local_rules.xml`

```
<group name="authentik,">
  <rule id="100200" level="0">
    <decoded_as>authentik</decoded_as>
    <description>Authentik event detected.</description>
  </rule>

  <rule id="100201" level="3">
    <if_sid>100200</if_sid>
    <match>login</match>
    <description>Authentik: User $(user) logged in successfully from $(srcip)</description>
    <group>authentication_success,</group>
  </rule>

  <rule id="100202" level="7">
    <if_sid>100200</if_sid>
    <match>login_failed|authorize_application_failed</match>
    <description>Authentik: Failed login attempt for user $(user) from $(srcip)</description>
    <group>authentication_failed,</group>
  </rule>
</group>
```

3. Verify and Restart

Always test the syntax before restarting, or the manager won't start back up.

```
# Test syntax
/var/ossec/bin/wazuh-analysisd -t

# If OK, restart
```

```
systemctl restart wazuh-manager
```

? How to Troubleshoot (The "Cheat Sheet")

If it stops working in the future, follow the data path:

1. **Is Docker sending logs?** `tail -f /var/log/syslog | grep authentik` (Run in Authentik LXC)
2. **Is the Manager receiving them?** `tail -f /var/ossec/logs/archives/archives.log | grep authentik` (Run in Wazuh Manager - requires `logall` enabled in `ossec.conf`)
3. **Is the logic correct?** Paste a log line into `/var/ossec/bin/wazuh-logtest` and look for **"Alert to be generated"**.

> **Note:** We also fixed the "Vulnerability Detector" noise by adding a suppression rule (ID 100060) in your `local_rules.xml` to hide those "Kernel vulnerability resolved" alerts!

Revision #1
Created 11 May 2026 01:53:53 by Christian Kaba
Updated 11 May 2026 02:05:27 by Christian Kaba